**REFramework Architecture**

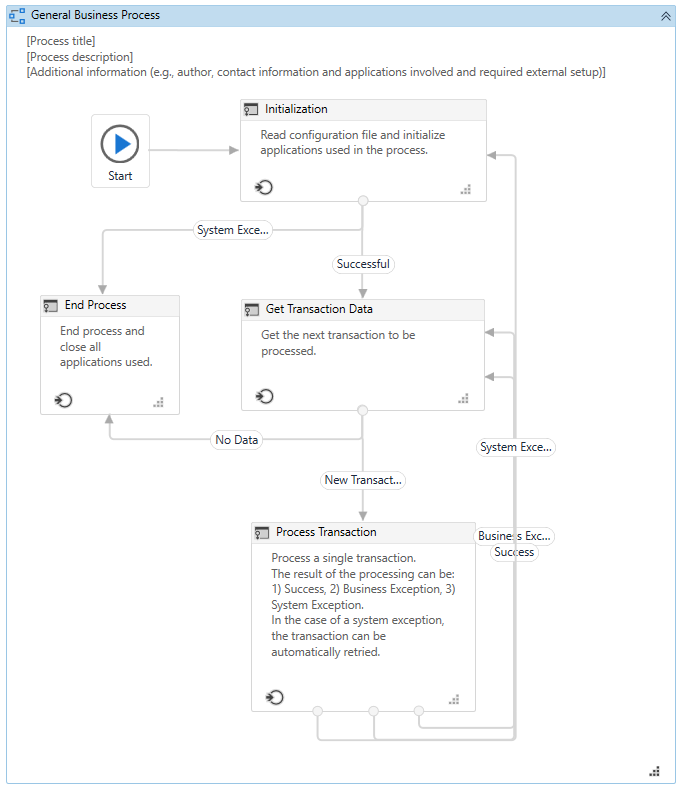
The REFramework is implemented as a state machine workflow, which is a kind of workflow that defines states that represent a particular circumstance of the execution. Depending on certain conditions, the execution can transition from one state to another to represent the steps of a process. States The states of the REFramework can be seen in Figure 6, and they are detailed as follows:

• **Initialization:** Read configuration file and initialize applications used in the process. If the initialization is successful, the execution moves to the Get Transaction Data state; in case of failure, it moves to the End Process state. If a system exception occurs during the processing of a transaction, the framework attempts to recover from the error by closing all applications used and returning to the Initialization state so the applications can be initialized again.

• **Get Transaction Data:** Get the next transaction to be processed. If there are no data to be processed or any errors occur, the execution goes to the End Process state. If a new transaction is successfully retrieved, it is processed in the Process Transaction state.

• **Process Transaction:** Process a single transaction. The result of the processing can be Success, Business Exception or System Exception. In the case of System Exception, the processing of the current transaction can be automatically retried. If the result is Business Exception, the transaction is skipped, and the framework tries to retrieve a new transaction in the Get Transaction Data state. The execution also returns to the Get Transaction Data state to retrieve a new transaction if the processing of the current one is successful.

• **End Process:** Finalize the process and close all applications used.

 State Machine with the States of the REFramework

**Exception Handling and Recovery**

The REFramework offers a robust exception handling scheme and can automatically recover from failures, update statuses of transactions and gracefully end the execution in case of unrecoverable exceptions. This feature is closely related to the logging capabilities, so that all information about exceptions is properly logged and available for analysis and investigation. Exceptions that happen during the framework’s execution are divided in two categories:

• **Business Exceptions:** This kind of exception is implemented by the class BusinessRuleException and it should be thrown when there are problems related to rules of the business process being automated. For example, if a process expects to receive an email with an attachment, but the attachment does not exist, the process would not be able to continue. In this case, a developer can use the Throw activity to throw a BusinessRuleException, which indicates that there was a problem that prevented the rules of the process to be followed. Note that BusinessRuleExceptions must be explicitly thrown by the developer of the workflow, and they are not automatically thrown by the framework or activities.

• **System Exceptions:** If an exception is not related to rules of the process itself, it is considered a system exception. Examples of system exceptions include an activity that timed-out due to slow network connection or a selector not found because of a browser crash. Depending on the category of exception, business exception or system exception, the REFramework decides whether the transaction should be retried. In the case of business exceptions, the transaction is not automatically retried, since issues related to business rules usually require human intervention. On the other hand, in the case of system exceptions, the error might have been caused by a temporary problem and retrying the same transaction can make it succeed without human intervention. Note that both business exceptions and system exceptions are concepts that also exist in Orchestrator under the names Business Exceptions and Application Exceptions. In fact, if the source of transactions is an Orchestrator queue, then the number of retries in the case of system exceptions can be set directly on Orchestrator. If Orchestrator is not used, the configuration for retries is done in the Config.xlsx file.